Identification and analysis of email and contacts artefacts on iOS and OS X

Ovens, Kenneth M.; Morison, Gordon

Published in:
11th International Conference on Availability, Reliability and Security (ARES), 2016

DOI:
10.1109/ARES.2016.56

Publication date:
2016

Document Version
Author accepted manuscript

Identification and Analysis of Email and Contacts Artefacts on iOS and OS X

Kenneth M. Ovens and Gordon Morison
Computer, Communications and Interactive Systems
Glasgow Caledonian University
Glasgow, Scotland
Email:{kenneth.ovens, gordon.morison}@gcu.ac.uk

***Abstract*—Acquiring data from cloud storage services has become increasingly important to digital forensic investigations. As more providers offer greater online storage facilities and user data is synchronised across multiple devices, an abundance of data sources has become available to assist with forensic investigations. However, such data can only become evidence when there is a thorough understanding of the data dynamics between client devices and the cloud, and there are explanations for any variations. This paper documents and analyses the artefacts created by interactions between Apple's cloud service, email and contacts applications. An explanation of as to why some artefacts synchronised over the cloud do not have matching cryptographic hashes is offered, and the ability to establish email origin on a system of multiple devices sharing a single account is established.**

***Keywords*-Digital Forensics; Apple; cloud computing; iOS; smart phones;**

I. Introduction

As more companies offer greater online data storage facilities at little or no cost, acquiring data from cloud storage services has become increasingly important to digital forensic investigations. However, acquiring data from cloud services presents problems to investigators in comparison to traditional computer forensics [1]. The virtualisation of data and its distribution across multiple servers in multiple locations means that obtaining a physical hard drive for imaging is very unlikely, if not impossible.

This has led to researchers developing innovative methods to acquire data from cloud storage services [2], [3], [4]. Data is increasingly shared across a user's multiple devices through the cloud and this can be leveraged by investigators to obtain data from cloud storage services. For example, in 2014 when Apple launched OS X 10.10 for its computers and laptops, and iOS 8 for its mobile phones and tablets, great emphasis was placed on the *Continuity* feature that provided increased automatic synchronisation of data between its cloud storage service, iCloud, and users' mobile phones, tablets and computers [5]. This automatic replication of data across client devices, as well as cloud storage provision, presents multiple sources of potentially valuable evidence for forensic investigators. However, such data can only become reliable evidence when there is a complete understanding of how the data is created, replicated, and distributed across client devices and online storage.

Currently there is still a lack of understanding of the relationship between data stored in the cloud and that retained on devices following an interaction [6]. These interactions between the cloud and client devices leave behind a potentially rich source of information, however evidentiary material may be misunderstood or even missed if the dynamics of cloud data is not established [7].

As data migrates from device to device with very little, if any, user interaction, it is important to be able to establish not only that the data exists, but also how it came to be on the device [8]. Furthermore, as investigators often need to link a suspect to a specific device at a specific time [9], it follows that there should be no doubt the device in question is indeed where the evidence originated from. To date there has been little research conducted into establishing from which device data originated, before being synchronised through the cloud and onto a users other devices.

Although there is a growing body of research into acquiring data from cloud storage services, there are still unanswered questions as to why certain recovered artefact hash values do not match the original. Changes to evidence should be avoided and any alterations must be explained [8]. Forensic investigators establish the integrity of evidence by comparing the cryptographic hash value of recovered artefacts with the original data. When a document is run through a hash function it produces a hash value. The slightest alteration to the document would produce a completely different hash value. Recent research in cloud forensics has produced mixed results regarding the integrity of recovered artefacts [3], [10].

To address these issues, the aim of this paper is to conduct research into the transfer of Apple's email and contacts data between a client's devices and the cloud, and by doing so answer the following questions:

1) Can the origin of email artefacts be determined in a system of multiple devices linked to a single account?
2) What causes a hash value mismatch for certain data across linked devices?

A. Contributions

This research will help investigators determine if suspect emails originated from the device being investigated. This can be crucial information required to link the suspect to the deed. An explanation is provided as to why emails synchronised across linked devices do not have matching hash values. This is important for explaining alterations to evidence. A further contribution of this paper is to provide

the forensic community with an understanding of the types and locations of email and contacts artefacts on iOS and OS X operating systems.

This paper is structured as follows: **Section Two** discusses the progress of research into data acquisition from cloud interactions. **Section Three** describes the experimental approach undertaken to address the research questions. **Section Four** reports the findings and discusses the results. **Section Five** draws conclusions from the research undertaken and presents future work.

II. Related Work

The problems facing cloud forensic investigators is well documented [11], [12], [13], [14] and there is now a growing body of research looking into one of the main issues, identifying and acquiring evidence from the cloud in a forensically sound manner.

A process for the recovery of artefacts that remained after accessing cloud storage services was devised [2]. This study also documented artefacts that can be found on computers (Windows, OS X) and smartphones (Android and iOS) that relate to usage of cloud services: Amazon S3, Dropbox, GoogleDocs and Evernote. A case study was also conducted in which files from System A were uploaded to Dropbox and subsequently located on System B after having been synchronised with the cloud server. Details of these artefacts were provided, as well as where they could be found on different operating systems, logs, databases and cache files. Although the artefacts were not hashed to confirm there were no alterations from the original files to synchronised versions, this study did explore the idea of data transitioning from one device to another via cloud storage services, which is very relevant.

Another study focused on three popular cloud storage providers: Dropbox, Box and SugarSync [6]. Rather than attempting to acquire the data stored on the cloud servers directly, the researchers proposed the use of smartphones as proxies to download the data. Using Android and iOS smartphones, the cloud services were accessed through the cloud storage applications that were downloaded and installed onto the smartphones. These applications create various artefacts that can be very useful to forensic investigations, including synchronised copies of the actual documents as well as associated metadata. It is not certain if the documents recovered were altered in any way from the original files, as the researchers did not appear to create cryptographic hashes for later comparison before uploading the data set to the cloud storage providers.

These studies established that data could be recovered from the cloud by linking devices to the account and synchronising the data to the device. Further research was required to determine if data copied from cloud storage services was an exact copy of the original. To put this another way: *"Does the act of collection result in changes to the data or its metadata?"*[3]. To answer this question the researchers analysed Dropbox, Google Drive and Microsoft SkyDrive and utilised both MD5 and SHA1 cryptographic hashing to confirm whether any changes or not were made to the content of the files downloaded. Network traffic was captured using the packet analyser, Wireshark, in an attempt to view any plain text communication between client and server. It was found that files stored in Dropbox and then downloaded via a browser interface matched the hashes of the original files. In fact these results were replicated for Google Drive and Microsoft SkyDrive - data remains the same, but timestamps alter, generally to when they have been downloaded.

In contrast to this, a study which focused specifically on Apple's cloud service, iCloud, found that synchronising data across devices, altered that data for certain applications [10]. The researcher located artefacts created by linking to the iCloud service and determined, by the use of cryptographic hashes, whether the artefacts created were identical to those on the original machine. As well as variations in timestamps, there were also variations in the MD5 hash values of files generated from preinstalled applications, despite the textual content of the files remaining unchanged. The files that appeared to alter without explanation included emails, contacts and databases.

This section showed the growing body of work investigating methods to acquire data using alternative methods. Data is being synchronised across cloud storage services and onto users' multiple devices providing forensic investigators with new sources from which they can recover evidence. However, an investigator may also need to know from which device evidence originated and to date, there is very little research in this area, other than a study that aimed to determine the origin of instant messages [15].

The following sections outline how the origin of certain artefacts, shared across linked devices and cloud storage, can be determined. Furthermore, an explanation is offered as to how synchronised artefacts may alter and prevent cryptographic hash values from matching.

III. Methodology

The aim of this research is to identify, compare and analyse email and address book artefacts on iOS 8 and OS X 10.10 devices. To achieve this, experiments were conducted involving the exchange of emails and contact data on an Apple iPad Mini and an Apple MacBook Pro. The experiments were broken into six stages. These stages included:

1) preparing the devices and creating a user account
2) creating email and contact artefacts based on typical user usage
3) documenting artefact metadata and hashes
4) capturing communications from each device using Wireshark
5) locating and documenting new and amended artefacts created by the usage of Apple's *Mail* and *Contacts* applications
6) comparing the new data on each device with the original data

Wireshark was deployed to capture the network traffic during each experiment. The resulting .PCAP file was later

analysed to determine the communication process between the devices and the cloud.

The iOS device used in the experiment is an iPad Mini model A1432 with iOS 8.1.2 installed. The iPad had been jailbroken using Pangu v1.2.1.[16]. Jailbreaking is a method developed in order to bypass Apple's access restrictions. It is a process that allows users to install and execute applications that have not been approved by Apple. These applications were required to access the iPad with a fully interactive shell, allowing the file system to be accessed and the artefacts hashed and analysed. Jailbreaking a device alters data that is stored on the device and as such this should not be taken as a recommended acquisition method. As the jailbreaking was done before the experiments were conducted, it will have had no adverse effect on the artefacts under discussion. Forensic investigators should acquire data from iOS devices using established acceptable tools and methods. For example, commercial tools such as Lantern by Katana Forensics, Forensic Extractor by Oxygen Forensics, and the Elcomsoft Mobile Forensic Bundle can acquire data from iOS devices depending on model types and versions. iTunes backup files could also be used recover data.

The OS X device used in the communication exchange was a virtual machine freshly installed with OS X 10.10.2. The use of a virtual machine was chosen as it allowed an efficient method of running multiple experiments by reverting the operating system and applications back to the original settings. This method has been used in previous research [10]. Specifications of the hardware for each device is detailed in Table I. These devices were chosen based on practicality and availability.

Table I
APPARATUS

Feature	iPad Mini (A1432)	MacBook Pro
Operating System	iOS 8.1.2 (12B440)	10.10.2
Processor	A5	2.6 GHz Intel Core i5
Memory	512MB	8GB 1600 MHz DDR3

The following steps were undertaken to prepare the iPad Mini for the experiments.

1) The iPad Mini was reset to its default factory settings.
2) A new user was created which creates a new Apple ID and iCloud account.
3) The operating system was updated to the latest version.
4) The iPad was connected via WiFi to the Internet.
5) The iPad was logged onto the iCloud account and configured to synchronise the following services: iCloud Drive, which included Safari, Photos, Mail, Contacts, Calendars, Reminders, Notes, Keychain and Find My iPad. The Mail account was configured with the new user account and Share My Location was activated.
6) New, fictional contacts were created using the *Contacts* application.
7) Emails were sent to and from a third-party email account and the iOS iPad Mini.

After email and contact data were created the relevant iOS artefacts were located immediately by connecting to the iPad via a secure shell (*SSH*) connection to search for files that had been recently modified using a *Unix* command which is shown and explained in Fig. 1.

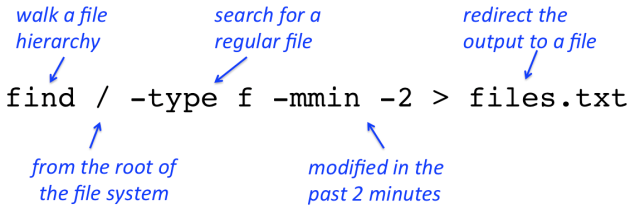

Figure 1. Breakdown of command used to find modified files.

The `find` command checks every file on the operating system and notes when it was last modified. A list is generated of those files that have been modified within the stated time. The list of files was then manually analysed by the researcher to separate relevant email and contact artefacts from unconnected background processes that were also present.

The following steps were undertaken to prepare the MacBook Pro for the experiment:

1) VMWare Fusion Professional (Version 7.1.1) was installed to allow for the creation of virtual machines.
2) Mac OS X 10.10.2 was installed in a virtual machine and updated.
3) A snapshot was taken of the operating system at this time to allow the virtual machine to be reset to this state for future experiments. A copy was made of the virtual machine image to reflect the state of the system before logging into the newly created user iCloud/Apple ID account.
4) From the iCloud application in System Preferences, the virtual machine was configured with the same Apple ID and user account as was configured for the iPad mini. This initiated synchronisation between the iCloud account and the MacBook Pro.
5) New, fictional contacts were created using the *Contacts* application.
6) Emails were sent to and from a third-party email account and the Mac OS X.
7) *Mail* and *Contact* applications were opened and closed to ensure documents were synchronised.

The relevant OS X artefacts were located by opening a terminal and searching for files that had been recently modified using the same *Unix* command as was used with the iPad.

Both devices were connected to a wireless access point that was being monitored by Wireshark. This allowed the capture and analysis of the communications between the client devices and the iCloud servers.

IV. Results and Analysis

This section reviews the results of the data analysis conducted after iOS, OS X *Mail* and *Contacts* usage. A description of *Mail's* communication process is presented first, followed by details of relevant artefact locations that remain after using *Mail*. The hash values of artefacts are then compared across OS X and iOS devices concluding with an analysis of the results. The same format is then used for the *Contacts* application.

A. Mail

1) Communication Process: Apple's *Mail* application uses the Internet Message Access Protocol over SSL (IMAPS), allowing multiple clients to simultaneously connect to the same mailboxes to retrieve emails. Devices frequently contact the Apple servers to check for any updates as well as updating the server of any local changes. Any changes made by one device will be replicated across all devices through the central point of the Apple mail servers. Simple Mail Transfer Protocol (SMTP) is used for sending emails. Table II shows the output of the Wireshark packet capture confirming the mail protocol used by iOS and OS X to retrieve messages.

2) File Locations: A relevant selection of artefacts, recently created and modified after an email exchange on OS X, is shown in Table III. The binary property list files mainly contain user preference settings. The log files didn't reveal any personal information of the user and no information could be extracted from the data files. Most email content and metadata is stored in the SQLite database - `Protected Index`. Emails are stored in folders relating to individual mail accounts (e.g. Gmail, Yahoo!, Hotmail), subdivided into mailbox folders: drafts, inbox, outbox, etc. Personal folders created by the user also show within the mail account folder. Individual emails (`.emlx` files) are located within their individual mailbox folders.

iOS mainly relies on SQLite databases to store emails and associated metadata. The two main databases used for this purpose are `Envelope Index` and `Protected Index`, shown in Table IV. Email attachments are embedded within the `message_data` table of the `Protected Index` table. The only instance observed of iOS saving an individual email file (`.emlx`), was when the email had been sent from the iOS device. The `Recents` database contained email addresses, paths of iCloud mailboxes and the map location of an Apple email.

3) Cryptographic Hash Values: Hash values were taken of an email file (`.emlx`) sent from the iOS device and the equivalent email file on the OS X device. Viewing the content of the stored files revealed that the OS X `.emlx` file had extra metadata embedded into the message that was not in the iOS version. This is clearly why these file hash values will not match.

4) Analysis: Users would see no discernible difference in an email on their iOS device compared to the same email on their OS X device. However, a forensic investigator would discover different cryptographic hash values of the underlying data from each source. This is because SQLite databases are unlikely to be exactly the same on different devices, especially if the devices use different operating systems.

Even if the database structures are identical, data is not always entered into the main database at the same time. The mechanism for entering data into SQLite databases initially involves a separate file, the Write Ahead Log (`.wal` file), which logs entries and only commits data to the main database file at certain checkpoints.

On occasions when there are individual email files that can be compared, for example, when the iOS device keeps an `.emlx` copy of an email sent locally, OS X embeds metadata within its copy of the file which will result in mismatched hashes.

While failing to match hashes may be problematic for forensic investigators, the subtle differences that can be found across devices and operating systems may prove to be valuable in discerning from which device artefacts originated. These experiments revealed that the iOS device will only keep `.emlx` copies of emails that originated locally. Furthermore, part of the metadata embedded within the OS X `.emlx` email file, shown in Listing 1, only appears when the email did not originate from the local device. This knowledge can useful to an investigation in determining how evidence came to be on a device.

```
<key>original-mailbox</key>
<string>imap://<username>@mail.mac.com/
    ↪ Sent%20Messages</string>
```

Listing 1. Metadata contained within the `.emlx` file

B. Contacts

1) Communication process: Based on the analysis of artefacts found on each device and packet data captured using Wireshark, it appears that after a *Contacts* file has been altered on the local device a communication is initiated with iCloud servers. A *vCard* is created and stored online as a `.vcf` file which is an Internet standard for the sharing of contact information. The standardisation of this format enables the transition of data from differing applications and operating systems. A user's other iCloud linked devices will then contact the relevant iCloud server to extract the information required and adapt it to the file structure of that device, updating the local database to include the new details. This communication process was captured using Wireshark and the data is shown in Table V.

2) File Locations: The results of querying the OS X file system after *Contacts* usage determined the locations of the artefacts relating to the *Contacts* application. OS X devices store most *Contacts* data in: `/Users/<username>/Library/ApplicationSupport/AddressBook/` as shown in Table VI. Local contacts are located in the SQLite database `AddressBook-v22.abcddb`, whereas online contacts (iCloud, MSExchange, etc.) are located in databases of the same name, but separated into individual

Table II
WIRESHARK CAPTURE OF *Mail* ICLOUD COMMUNICATION USING IMAP OVER SSL

No.	Source	Destination	Protocol	Length	Info
1	192.168.100.55	192.168.100.1	DNS	91	Standard query 0xa433 A p33-imap.mail.me.com.akadns.net
2	192.168.100.1	192.168.100.55	DNS	107	Standard query response 0xa433 A 17.142.165.10
3	192.168.100.55	17.142.165.10	TCP	78	53829 >**imaps** [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=987360204 TSecr=0 SACK_PERM=1
4	17.142.165.10	192.168.100.55	TCP	74	**imaps** >53829 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1380 SACK_PERM=1 TSval=679404437 TSecr=987360204 WS=128
5	192.168.100.55	17.142.165.10	TCP	66	53829 >**imaps** [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=987360392 TSecr=679404437
6	192.168.100.55	17.142.165.10	TLSv1	237	Client Hello

Table III
OS X *Mail* ARTEFACT LOCATION AND TYPE

File Location	File Type
`/Users/<username>/Library/Containers/com.apple.mail/Data/Library/Logs/Mail/2015-03-30_AccountManager.log`	Debug log
`/Users/<username>/Library/Containers/com.apple.mail/Data/Library/Preferences/com.apple.mail.plist`	Binary property list
`/Users/<username>/Library/Containers/com.apple.mail/Data/Library/SavedApplicationState/com.apple.mail.savedState/data.data`	Data
`/Users/<username>/Library/Mail/V2/IMAP-<username>@mail.com@imap.mail.com/INBOX.mbox/<UUID>/Data/Messages/19.emlx`	Email message
`/Users/<username>/Library/Mail/V2/IMAP-<username>@mail.com@imap.mail.com/INBOX.mbox/Info.plist`	Binary property list
`/Users/<username>/Library/Mail/V2/MailData/EnvelopeIndex-wal`	SQLite database log

Table IV
IOS *Mail* ARTEFACT LOCATION AND TYPE

File Location	File Type
`/private/var/mobile/Library/Mail/EnvelopeIndex-wal`	SQLite database log
`/private/var/mobile/Library/Mail/ProtectedIndex-wal`	SQLite database log
`/private/var/mobile/Library/Mail/Recents`	SQLite database
`/private/var/mobile/Library/Mail/iCloud-<username>/.mboxCache.plist`	Binary property list
`/private/var/mobile/Library/Mail/metadata.plist`	Binary property list

folders within the `Sources` folder. A record of every new contact, change and deletion is recorded in another database `ABAssistantChangelog.aclcddb`.

OS X devices also store individual card files for each contact and group in the `Metadata` folder in the form of a binary property list. Profile pictures are stored in the `Images` folder. Files with `-shm` and `-wal` extensions are components of the SQLite database of the same name.

For iOS devices, artefacts created or modified after recent *Contacts* usage were located in `/private/var/mobile/Library/`, the details of which are shown in Table VII. *Contacts* profile images are stored in `/private/var/mobile/Library/AddressBookImages.sqlitedb`.

As well as the standard contact details that would be expected within an address book application, there are other revealing details within the `AddressBook.sqlitedb` (iOS) and `AddressBook-v22.abcddb` (OS X) databases. Uniform Resource Locators (URLs) can be extracted from the database specifying the location of both profile images and `.vcf` files stored in the iCloud account. Using the command line tool `cURL`, it is possible to directly connect to the iCloud server, located at `https://p33-contacts.icloud.com:443`, and download these files to any computer with no requirement for an OS X or iOS device.

```
curl -u <email account>:<password> -Ov
    ↪ https://<email account>@p33-
    ↪ contacts.icloud.com/<External ID
    ↪ >/carddavhome/card/<GUID>.vcf
```

Listing 2. `cURL` command used to download contact card

The URL location is documented in the `ABPerson` table within the iOS SQLite database `AddressBook.sqlitedb` and in the `ZABCDCUSTOMPROPERTYVALUE` table within the OS X `AddressBook-v22.abcddb` database.

3) Cryptographic Hash Values: As shown in the previous sections, iOS and OS X devices store data from the *Contacts* applications in different ways. iOS devices store the majority of data in the SQLite database `AddressBook.sqlitedb`, with profile pictures stored in `AddressBookImages.sqlitedb`. Whereas OS X devices have a database (`AddressBook-v22.abcddb`) for some data, another database for indexing any changes (`ABAssistantChangelog.aclcddb`), as well as separate folders for profile pictures and individual contact files. There appears to be no comparative artefact that can be used for hashing comparison.

4) Analysis: From a user's perspective, a contact on an iOS iPad and the same contact on an OS X MacBookPro will look the same and contain the same details. However, the way in which each operating system stores this information differs. While both systems make use of

Table V
WIRESHARK CAPTURE OF *Contacts* ICLOUD COMMUNICATION

No.	Source	Destination	Protocol	Length	Info
1	192.168.100.55	192.168.100.1	DNS	94	Standard query 0xf316 A p33-contacts.icloud.com.akadns.net
2	192.168.100.1	192.168.100.55	DNS	110	Standard query response 0xf316 A 17.142.164.22
3	192.168.100.55	17.142.164.22	TCP	78	53761 >https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=968316153 TSecr=0 SACK_PERM=1
4	17.142.164.22	192.168.100.55	TCP	58	https >53761 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1360
5	192.168.100.55	17.142.164.22	TCP	54	53761 >https [ACK] Seq=1 Ack=1 Win=65535 Len=0
6	192.168.100.55	17.142.164.22	TLSv1.2	280	Client Hello

Table VI
OS X *Contacts* ARTEFACT LOCATION AND TYPE

File Location	File Type
/Users/<username>/Library/ApplicationSupport/AddressBook/ABAssistantChangelog.aclcddb-shm	SQLite shared memory
/Users/<username>/Library/ApplicationSupport/AddressBook/ABAssistantChangelog.aclcddb-wal	SQLite database log
/Users/<username>/Library/ApplicationSupport/AddressBook/Metadata/.info	Binary property list
/Users/<username>/Library/ApplicationSupport/AddressBook/Sources/<UUID>/AddressBook-v22.abcddb-wal	SQLite database log
/Users/<username>/Library/ApplicationSupport/AddressBook/Sources/<UUID>/Metadata/.info	Binary property list
/Users/<username>/Library/ApplicationSupport/AddressBook/Sources/<UUID>/Metadata/1F386650-3419-4083-B07F-8DC3376A888E:ABGroup.abcdg	Binary property list
/Users/<username>/Library/ApplicationSupport/AddressBook/Sources/<UUID>/Metadata/9CD94F50-D1DA-44D1-8F89-7D782897346F:ABPerson.abcdp	Binary property list
/Users/<username>/Library/ApplicationSupport/AddressBook/Sources/<UUID>/Images/6B4B1DDD-F94A-4897-A06A-EB3933C5B513	JPEG image data
/Users/<username>/Library/ApplicationSupport/AddressBook/Sources/<UUID>/Metadata/CED7FC41-CA62-4571-A494-00DA0AD63492:ABInfo.abcdi	Binary property list
/Users/<username>/Library/Caches/com.apple.AddressBookSourceSync/Cache.db-shm	SQLite shared memory

Table VII
IOS *Contacts* ARTEFACT LOCATION AND TYPE

File Location	File Type
/private/var/mobile/Library/AddressBook.sqlitedb-wal	SQLite database log
/private/var/mobile/Library/AddressBookImages.sqlitedb-wal	SQLite database log

SQLite databases, OS X also keeps individual contact files and profile images in separate folders. For iOS devices, everything is embedded within databases. Further, the iOS and OS X databases differ in the number of tables, naming conventions and table structures. Yet both systems are able to display and share the same content. This is made possible by using a standardised format for storing the data in the iCloud. There were no **vCards** found on either the OS X or iOS system, but this appears to be how the information is stored centrally and accessed by each device.

V. CONCLUSION AND FUTURE WORK

User data is increasingly shared across a user's multiple devices and on cloud storage platforms. As data migrates from device to device and through the cloud, it is important for forensic investigators to be able to establish not only where evidence can be located, but also how it came to be there.

This paper presented an analysis of email and contacts artefacts on iOS and OS X devices. The results presented in the previous sections, provide answers to the research questions posed in Section One.

1) This study has been able to show subtle differences in artefacts that can help determine the origin of certain email communications.
2) An explanation has been provided as to why certain data stored on different devices can not have matching hash values.

This study has also documented the type, location and content of artefacts that could be helpful in forensic investigations. Understanding the origin of these artefacts could help determine whether the evidence recovered is incriminating or exculpatory.

This study was limited to Apple devices, *Mail* and *Contacts* applications. Additional research would be required to determine if similar results would occur in third-party mail providers such as Hotmail and Gmail. The scope of this research can also be widened as personal data is shared between more devices with the introduction of watches, fitness trackers, televisions and domestic appliances.

REFERENCES

[1] NIST Cloud Computing Forensic Science Working Group, "NIST Cloud Computing Forensic Science Challenges (Draft)," National Institute of Standards and Technology, Tech. Rep., 2014. [Online]. Available: http://csrc.nist.gov/publications/drafts/nistir-8006/draft_nistir_8006.pdf

[2] H. Chung, J. Park, S. Lee, and C. Kang, "Digital forensic investigation of cloud storage services," *Digital*

Investigation, vol. 9, no. 2, pp. 81–95, Nov. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287612000400

[3] D. Quick, B. Martini, and K.-K. R. Choo, *Cloud Storage Forensics*, 1st ed. Syngress, 2014.

[4] F. Daryabar, A. Dehghantanha, B. Eterovicsoric, and K.-K. R. Choo, "Forensic investigation of OneDrive , Box , GoogleDrive and Dropbox applications on Android and iOS devices," *Australian Journal of Forensic Sciences*, vol. 0618, Mar. 2016.

[5] Apple Inc., "Mac and iOS. They really click." 2014. [Online]. Available: https://www.apple.com/osx/continuity

[6] G. Grispos, W. B. Glisson, and T. Storer, "Using smartphones as a proxy for forensic evidence contained in cloud storage services," *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 4910–4919, 2013.

[7] M. Govan, "Forensic droplets & puddles from the cloud," in *Cyberforensics Perspectives*, G. R. S. Weir and M. Daley, Eds. Cardiff: University of Strathclyde Publishing, 2013, pp. 11–17.

[8] J. Williams, "ACPO Good Practice Guide for Digital Evidence," Association of Chief Police Officers, Tech. Rep. March, 2012. [Online]. Available: http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf

[9] B. Shavers, "Foreword," in *Placing the Suspect Behind the Keyboard*. Boston: Syngress, 2013, pp. xvii – xix. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781597499859000198

[10] K. Oestreicher, "A forensically robust method for acquisition of iCloud data," *Digital Investigation*, vol. 11, pp. S106–S113, 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1742287614000498

[11] D. Birk and C. Wegener, "Technical issues of forensic investigations in cloud computing environments," in *Systematic Approaches to Digital Forensic Engineering (SADFE), 2011 IEEE Sixth International Workshop on*, May 2011, pp. 1–10.

[12] K. Ruan, J. Carthy, T. Kechadi, and I. Baggili, "Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results," *Digital Investigation*, vol. 10, no. 1, pp. 34–43, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2013.02.004

[13] S. O'Shaughnessy and A. Keane, "Impact of cloud computing on digital forensic investigations," in *Advances in Digital Forensics IX*, ser. IFIP Advances in Information and Communication Technology, G. Peterson and S. Shenoi, Eds. Springer Berlin Heidelberg, 2013, vol. 410, pp. 291–303.

[14] S. L. Garfinkel, "Digital forensics research: The next 10 years," *Digital Investigation*, vol. 7, no. SUPPL., pp. S64–S73, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2010.05.009

[15] M. Govan and K. M. Ovens, "Temporal analysis anomalies with iOS iMessage communication exchange," in *Cyberforensics*, G. Weir, Ed. Glasgow: University of Strathclyde Publishing, 2014, pp. 37–49.

[16] Team Pangu, "Pangu ios 8 jailbreak tool," http://en.pangu.io/.